CLAIMS

We claim:

1. A system for interdicting unauthorized copying in a decentralized network comprising:

a plurality of software agents masquerading as nodes in a decentralized network; and

a query matcher that receives search results from the plurality of software agents, and reports matches of the search results with protected files back to the plurality of software agents so that the software agents can interdict unauthorized copying of the protected files in the decentralized network.

2. The system according to claim 1, wherein the plurality of software agents reside on one or more computers while communicating to the decentralized network through individually assigned ports.

3. The system according to claim 2, wherein the assigned ports have corresponding IP addresses that change in a manner so that detection of the plurality of software agents as unauthorized masqueraders of nodes in the decentralized network is made difficult.

4. The system according to claim 2, wherein the number and geographical locations of the one or more computers is determined by the number and geographical distribution of nodes in the decentralized network.

5. The system according to claim 1, wherein the query matcher has a database including metadata for the protected files.

6. The system according to claim 1, further comprising a central coordinating authority coordinating activities of the plurality of software agents so as to interdict unauthorized copying in the decentralized network.

7. The system according to claim 6, wherein the central coordinating authority sends instructions to the plurality of software agents specifying actions to be taken when the plurality of software agents receive matches of the search results with protected files back from the query matcher.

8. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to generate modified search results by deleting at least a subset of references corresponding to the matches of the search results, and forward the modified search results through the decentralized network.

9. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to generate modified search results by modifying at least a subset of references corresponding to the matches of the search results so as to point to one or more IP addresses that are invalid, and forward the modified search results through the decentralized network.

10. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to generate modified search results by modifying at least a subset of references corresponding to the matches of the search results so as to point to one or more IP addresses of nodes that do not have copies of the subset of references, and forward the modified search results through the decentralized network.

11. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to generate modified search results by modifying at least a subset of references corresponding to the matches of the search results so as to point to one or more IP addresses of nodes that are not connected to the decentralized network, and forward the modified search results through the decentralized network.

12. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to generate modified search results by modifying at least a subset of references corresponding to the matches of the search results so as to point to alternative files, and forward the modified search results through the decentralized network.

13. The system according to claim 12, wherein the alternative files include at least one randomly selected file residing on a node upon which one of the matches of the search results resides.

14. The system according to claim 12, wherein the alternative files include at least one decoy file residing on a host node controlled by the central coordinating authority.

15. The system according to claim 12, wherein the alternative files include at least one randomly selected file residing on a host node controlled by the central coordinating authority.

16. The system according to claim 12, wherein the alternative files include at least one rights-managed version of the matches.

17. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to send an alternative file to a client node when a request for a protected file is received from the client node.

18. The system according to claim 17, wherein the alternative file is a decoy.

19. The system according to claim 18, wherein the decoy is an audio file containing white noise.

20. The system according to claim 18, wherein the decoy is a video file containing white noise.

21. The system according to claim 18, wherein the decoy is an application containing a NOP executable that terminates the application when executed.

22. The system according to claim 18, wherein the decoy is an image file containing snow.

23. The system according to claim 18, wherein the decoy is a document with blank contents.

24. The system according to claim 18, wherein the decoy contains an anti-piracy message.

25. The system according to claim 17, wherein the alternative file is a rights managed version of the protected file.

26. The system according to claim 17, wherein the instructions sent by the central coordinating authority include an instruction to transmit the alternative file such that the transmission rate slows down during the transmission.

27. The system according to claim 17, wherein the instructions sent by the central coordinating authority include an instruction to transmit the alternative file such that the transmission terminates automatically after most, but not all of the alternative file has been downloaded.

28. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to modify at least one reference corresponding to a match in the search results so as to point to a non-existent file along with a reported hash

value that does not correspond to any file in the decentralized network instead of the at least one reference.

29. The system according to claim 7, wherein the instructions sent by the central coordinating authority include an instruction to modify a reference corresponding to a match in the search results so as to point to a spoof file instead of the reference and report a hash value matching that of the reference even though the contents of the spoof file do not exactly match that of the reference.

30. The system according to claim 1, wherein the decentralized network comprises:

a plurality of nodes; and

a plurality of supernodes individually having higher resources than each of the plurality of nodes so that a search string initiated from one of the plurality of nodes is first routed to one of the plurality of supernodes.

31. The system according to claim 30, wherein the plurality of software agents inform their respective supernodes that they have copies of protected files and claim node attributes so that the plurality of software agents will be selected as top matches by their respective supernodes for search strings indicating the protected files.

32. The system according to claim 30, wherein the plurality of software agents inform the decentralized network that they are supernodes.

33. The system according to claim 30, wherein the plurality of software agents report to the decentralized network that they possess attributes that qualify them as supernodes under the protocol of the decentralized network.

34. A method for interdicting unauthorized copying in a decentralized network, comprising:

infiltrating a decentralized network with a plurality of software agents masquerading as nodes so as to intercept communications related to search queries;

identifying references to protected files in the communications; and

interdicting unauthorized copying of the protected files with respect to the communications.

35. The method according to claim 34, wherein the decentralized network is an hierarchical network with supernodes and regular nodes, and the plurality of software agents masquerade as regular nodes that inform their respective supernodes that they have copies of protected files and claim node attributes so that the plurality of software agents will be selected as top matches by their respective supernodes for search strings indicating the protected files.

36. The method according to claim 34, wherein the decentralized network is an hierarchical network with supernodes and regular nodes, and the plurality of software agents inform the decentralized network that they are

supernodes according to the protocol of the decentralized network.

37. The method according to claim 34, wherein the decentralized network is an hierarchical network with supernodes and regular nodes, and the plurality of software agents report to the decentralized network that they possess attributes that qualify them as supernodes under the protocol of the decentralized network.

38. The method according to claim 34, wherein the communications are search results, and the interdicting of unauthorized copying comprises: generating modified search results by deleting at least a subset of references corresponding to the protected files in the search results, and forwarding the modified search results through the decentralized network.

39. The method according to claim 34, wherein the communications are search results, and the interdicting of unauthorized copying comprises: generating modified search results by modifying at least a subset of references corresponding to the protected files in the search results to point to one or more invalid IP addresses, and forwarding the modified search results through the decentralized network.

40. The method according to claim 34, wherein the communications are search results, and the interdicting of unauthorized copying comprises: generating modified search results by modifying at least a subset of references corresponding to the protected files in the search results

to point to one or more IP addresses that do not host the subset of references, and forwarding the modified search results through the decentralized network.

41. The method according to claim 34, wherein the communications are search results, and the interdicting of unauthorized copying comprises: generating modified search results by modifying at least a subset of references corresponding to the protected files in the search results to point to one or more IP addresses that are not connected to the decentralized network, and forwarding the modified search results through the decentralized network.

42. The method according to claim 34, wherein the communications are search results, and the interdicting of unauthorized copying comprises: generating modified search results by modifying at least a subset of references corresponding to the protected files in the search results to point to alternative files, and forwarding the modified search results through the decentralized network.

43. The method according to claim 42, wherein the alternative files include at least one synthesized decoy file.

44. The method according to claim 42, wherein the alternative files include at least one rights-managed version of one of the protected files referenced in the search results.

45. The method according to claim 34, wherein one of the communications is a request from a client node

to one of the plurality of software agents for a copy of a protected file, and further comprising: sending an alternative file to the client node in lieu of the copy of the protected file.

46. The method according to claim 45, wherein the alternative file is a synthesized decoy file.

47. The method according to claim 46, further comprising: synthesizing the decoy file by filling the decoy file with white noise.

48. The method according to claim 46, further comprising: synthesizing the decoy file by filling the contents of the decoy file with an anti-piracy message.

49. The method according to claim 46, wherein the protected file is an application program, and further comprising: synthesizing the decoy file by including a NOP executable that terminates when executed.

50. The method according to claim 45, wherein the alternative file is a rights-managed version of the protected file.

51. The method according to claim 45, wherein the sending an alternative file comprises: transmitting the alternative file at a transmission rate that slows down during the transmission.

52. The method according to claim 45, wherein the sending an alternative file comprises: transmitting the

alternative file in a manner such that the transmission terminates automatically after most, but not all of the alternative file has been downloaded.

53. The method according to claim 34, wherein one of the communications is search results, and the interdicting of unauthorized copying comprises: generating modified search results by providing a pointer to a non-existent file instead of another pointer to a reference in the search results that matches a protected file, and forwarding the modified search results through the decentralized network.

54. The method according to claim 53, wherein a reported hash value that does not match any file in the decentralized network is provided along with the pointer to the non-existent file.

55. The method according to claim 34, wherein one of the communications is search results, and the interdicting of unauthorized copying comprises: generating modified search results by replacing a pointer to a reference in the search results that matches a protected file with another pointer to a spoof file along with a hash value matching that of the reference, and forwarding the modified search results through the decentralized network.

56. The method according to claim 34, wherein one of the communications is a request to one of the plurality of software agents from a client node for at least a segment of a protected file, and the interdicting

of unauthorized copying comprises: transmitting data to the client node in response to the request so that a corrupted file is detected upon completion of downloading of the protected file to the client node.

57. A method for interdicting unauthorized copying in a decentralized network, comprising: interposing one or more software agents resembling nodes between a client node and neighboring nodes of the client node in a decentralized network such that all communications related to search queries must pass through the one or more software agents so as to allow the one or more software agents to interdict unauthorized copying by the client node in the decentralized network.

58. The method according to claim 57, wherein each of the neighboring nodes is directly connected to the client node, and the interposing one or more software agents comprises connecting the one or more software agents to the client computer so as to cause the client node to disconnect from the neighboring nodes and only be directly connected to the one or more software agents.

59. The method according to claim 57, wherein each of the neighboring nodes is directly connected to the client node in the decentralized network, and the interposing one or more software agents comprises for each of the neighboring nodes: connecting a corresponding one of the one or more software agents to that neighboring node and to the client node; and causing that neighboring node to be disconnected from the client node.

60. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message to that neighboring node to disconnect from the client node.

61. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message to the client node to disconnect from that neighboring node.

62. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message to that neighboring node to disconnect from the decentralized network.

63. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message purported to be from that neighboring node to the client node indicating that that neighboring node is disconnecting from the client node.

64. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message purported to be from the client node to that neighboring node indicating that the client node is disconnecting from that neighboring node.

65. The method according to claim 59, wherein the causing that neighboring node to be disconnected from

the client node comprises: issuing a message to that neighboring node that violates an agreed upon protocol between the client node and that neighboring node so as to cause that neighboring node to abandon the connection with the client node.

66. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: issuing a message to the client node that violates an agreed upon protocol between the client node and that neighboring node so as to cause the client node to abandon the connection with that neighboring node.

67. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: connecting additional software agents resembling nodes to the client computer until the client computer disconnects from that neighboring node.

68. The method according to claim 67, wherein the connecting additional software agents comprises: causing the client computer to transfer a connection to that neighboring node to another neighboring node so as to no longer be directly connected to that neighboring node.

69. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: bombarding a socket connection connecting the client node to that neighboring node with communications so as to cause the socket connection to be terminated.

70. The method according to claim 59, wherein the causing that neighboring node to be disconnected from the client node comprises: causing software running on that neighboring node and responsible for maintaining a connection with the client node to experience a known defect causing that neighboring node to be disconnected from the client node.